Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

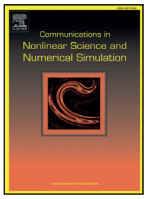
Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.

Commun Nonlinear Sci Numer Simulat 85 (2020) 105228

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns



Research paper

Topological data analysis to model the shape of immune responses during co-infections

Karin Sasaki[a], Dunja Bruder[d,e], Esteban A. Hernandez-Vargas[a,b,c,*]

[a] Frankfurt Institute for Advanced Studies, Frankfurt am Main 60438, Germany
[b] Instituto de Matematicas, UNAM, Unidad Juriquilla, Blvd. Juriquilla 3001, Queretaro C.P. 76230, Mexico
[c] Xidian-FIAS Joint Research Center, Germany-China
[d] Infection Immunology Group, Institute of Medical Microbiology, Infection Prevention and Control, Health Campus Immunology, Infectiology and Inflammation Otto-von-Guericke University Magdeburg, Germany
[e] Immune Regulation Group, Helmholtz Centre for Infection Research, Braunschweig, Germany

ARTICLE INFO

Article history:
Received 7 October 2019
Revised 17 January 2020
Accepted 11 February 2020
Available online 15 February 2020

Keywords:
Topological data analysis
Immune system dynamics
Influenza infections
Complex data analysis

ABSTRACT

Co-infections by multiple pathogens have important implications in many aspects of health, epidemiology and evolution. However, how to disentangle the non-linear dynamics of the immune response when two infections take place at the same time is largely unexplored. Using data sets of the immune response during influenza-pneumococcal co-infection in mice, we employ here topological data analysis to simplify and visualise high dimensional data sets.

We identified persistent shapes of the simplicial complexes of the data in the three infection scenarios: single viral infection, single bacterial infection, and co-infection. The immune response was found to be distinct for each of the infection scenarios and we uncovered that the immune response during the co-infection has three phases and two transition points. During the first phase, its dynamics is inherited from its response to the primary (viral) infection. The immune response has an early shift (few hours post co-infection) and then modulates its response to react against the secondary (bacterial) infection. Between 18 and 26 h post co-infection the nature of the immune response changes again and does no longer resembles either of the single infection scenarios.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Epidemics by infectious pathogens are a major threat to humankind. Viruses such as coronavirus, HIV, influenza, and ebola have caused the largest pandemics in the 20th century. A major problem is the simultaneous infection of a host by two or more pathogens (co-infection). We are continuously exposed to multiple potential pathogens; many people are chronically (e.g. HIV) or latently (e.g. herpes viruses) infected, and we all carry potential pathogens in our colonising microbial flora. This means that nearly every new infection is some sort of co-infection, and globally, co-infections are the norm rather than the exception [1].

There is an impressive number of combinations of pathogens that derive synergy from contemporaneous infection of a host. These include viral-bacterial (e.g. influenza with pneumococcus or HIV with Mycobacterium tuberculosis), viral-viral

* Corresponding author at: Frankfurt Institute for Advanced Studies, 60438 Frankfurt am Main, Germany.
E-mail address: esteban@im.unam.mx (E.A. Hernandez-Vargas).

https://doi.org/10.1016/j.cnsns.2020.105228
1007-5704/© 2020 Elsevier B.V. All rights reserved.

(e.g. Hepatitis B with Hepatitis C), bacterial-bacterial (e.g. Borrelia burgdorferi with Anaplasma phagocytophila in Tick-borne illnesses) and pathogen-pathogen (e.g. Malaria with Dengue, Chikungunya, Filaria or Helminth) co-infections, to name a few. Although most studies to date have been focused on co-infections between two pathogens, infections with multiple pathogens are now becoming active topics of research [2].

Co-infections have effects on health at multiple levels: Co-infections can increase or decrease the rate of transmission of other infections [3], modulate the host immune response [4], create protection and resilience or susceptibility to further infections [5,6], alter the performance of diagnostic tests and antimicrobial chemotherapy [7,8], and even create opportunities for the emergence of new pathogens [9,10]. In other words, some co-infections can have detrimental, or even beneficial, outcomes.

The harmful effects of chronic co-infections, such as tuberculosis or Hepatitis B and C in association with HIV for example, are well established. However, generally and especially in acute infections, the mechanisms of co-pathogens with the host immune system and the possible consequences, ranging from insignificant, harmful or beneficial, are still largely unknown and difficult to dissect. The development of mathematical approaches that characterise the immune responses in the host have offered important steps for studying pathogen - pathogen interactions [11]. Interpretation of data sets with modern mathematical and machine-learning strategies can provide a comprehensive understanding of co-infections and their relevance/significance.

Topological Data Analysis (TDA) is a collection of computational tools derived from the mathematical subject of Algebraic Topology, that can help in identifying the behaviour of a biological system from a global perspective, guide detailed quantitative investigations and aid tailor further experimental settings. In fact, algorithms from topological data analysis have started to play important roles in novel interdisciplinary fields in biomedical sciences, including cancer genomics [12], diabetes [13], neuroscience [14], infectious diseases [15,16], and in biology in general [17,18].

Among the different TDA techniques for the qualitative analysis, the mapper algorithm [19] has shown a potential to simplify and visualize high dimensional data sets. It generates a simple description of the data in the form of a combinatorial object called a *simplicial complex*, that captures topological and geometric information of the point cloud in high dimensional space. The algorithm uses a (combination of) function(s) that map the data to a metric space, and builds an informative representation based on the clustering of subsets (which are associated to the values of the function(s)) of the data set. In the simplest case, this method reduces high dimensional data sets to a network whose nodes correspond to clusters in the data and edges to the existence of points in common between clusters. The aim of this algorithm is not to obtain a fully accurate representation of a data set, but rather a low-dimensional image which can highlight areas of interest, possibly for further analysis and quantification.

Motivated by the obvious potential of topological investigations in biomedical sciences, in the present study we seek to understand the evolution of the immune system as it responds to co-infection between virus and bacteria. Mathematical modeling research in influenza-pneumococcal co-infections has been a growing field within last years [4,20–23]. These previous approaches are based on differential equations constructed based on biological reasoning. While they are suitable tools to test different hypothesis, and have helped elucidate many of the details of the mechanisms of these intricate systems, these models are susceptible to bias by the designer and model complexity rapidly limits the reliability in the parameter fitting procedures [24]. In contrast, TDA is a tool that detects true patterns in the data, without imposing artificial assumptions.

Here, we use the co-infection data sets from [4] where we investigated the hierarchical effects of pro-inflammatory cytokines on the post-influenza susceptibility to pneumococcal co-infection by assessing the early and late kinetics of pro-inflammatory cytokines in the respiratory tract. In the experimental part of this study mice were divided into three groups and given either a single viral infection (with IAV strain A/PR8/34), a single bacterial infection (S. pneumoniae strain T4) or a co-infection (IAV + T4). The experimental readouts were the bacterial burden, viral titers and cytokine concentrations in the lung. Our previous work [4] used mathematical modelling that suggested a detrimental role of IFN-γ alone and in synergism with IL-6 and TNF-α in impaired bacterial clearance. We now use the mapper algorithm to investigate the global shape of the immune response under the three above infection scenarios, illustrated in Fig. 1.

2. Topological data analysis

In this section we give a brief and intuitive introduction to the mapper algorithm and the algebraic topology concepts behind it. We also present in detail our methodology for using the mapper algorithm to investigate influenza in co-infection with bacteria. For further details, we refer the reader to the following resources: the original paper of the mapper algorithm is [19]; the paper presenting the Kepler-Mapper Python Library that we used to implement the mapper algorithm computationally is [25]; we also recommend [26] as an introductory book to computational algebraic topology (albeit it does not include the mapper algorithm) and [27] for a thorough treatment of the mathematical subject of algebraic topology.

2.1. The mapper algorithm

The mapper algorithm is a method of replacing a topological space by a simpler one, known as a *simplicial complex*, which captures topological and geometric features of the original space. The purpose of doing this is not only to obtain a visualization of high dimensional data sets in 3-d, but also because mathematical properties of simplicial complexes allow

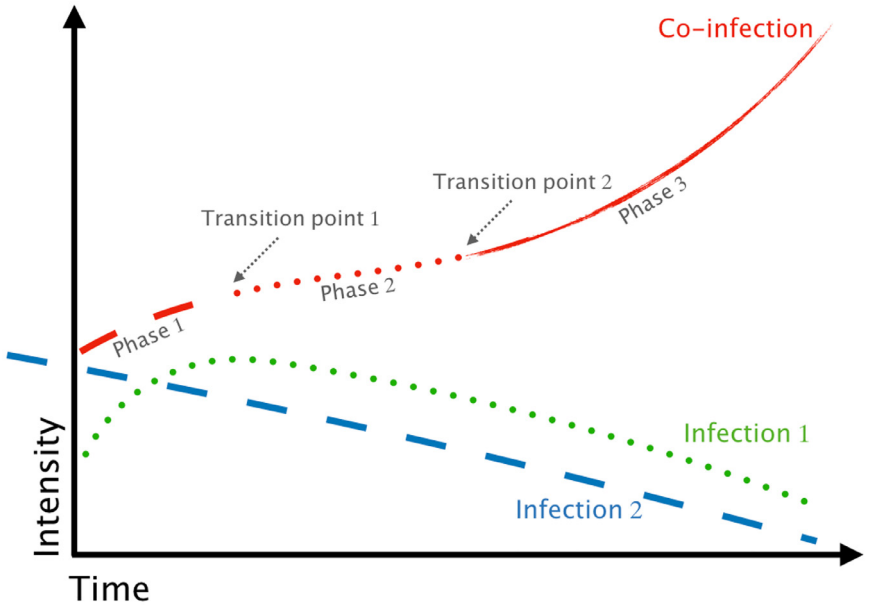


Fig. 1. Predictions about the behaviour of the immune system in response to co-pathogenesis in the lung. Topological data analysis and nearest neighbour analysis reveal that initially the immune system inherits its behaviour from its response to the primary infection; it goes through a swift transition early in the co-infection (i.e. soon after the onset of the secondary infection) and it is consequently and temporarily driven mainly by its response to the secondary infection. There is a second transition point in the behaviour of the immune system and from there, it no longer resembles a standard response associated to either of the two single infections, but rather it shapes its behaviour to respond to the co-infection itself.

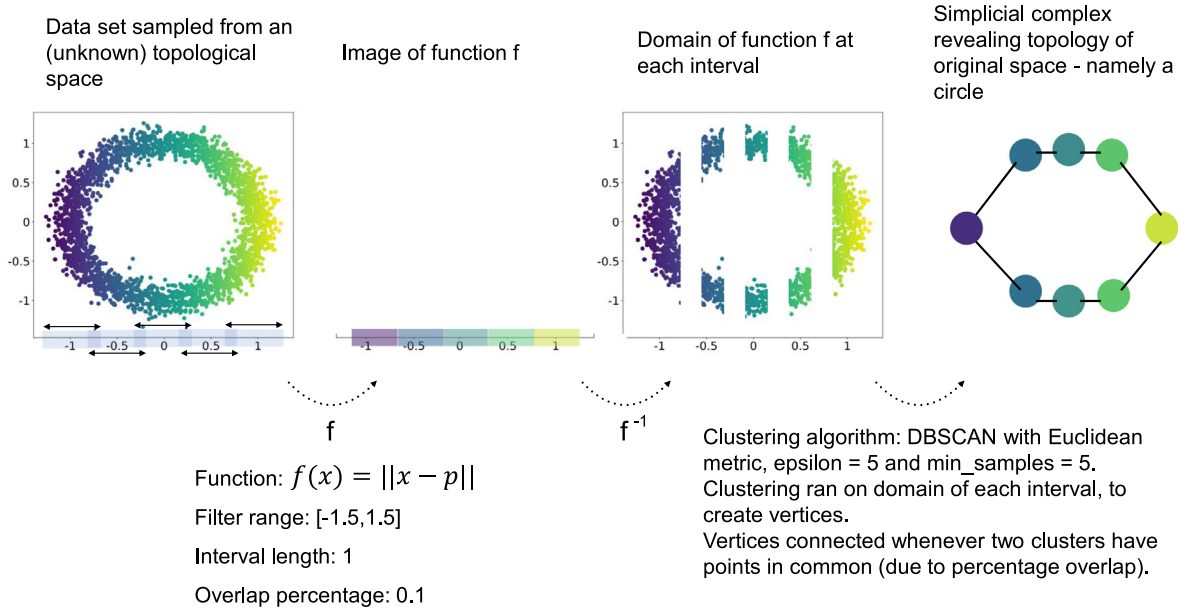


Fig. 2. Visualisation of the steps of the mapper algorithm being applied to a set of points sampled from the 2-dimensional circle. Parameter p in the function used as a lens is the leftmost point in the data. Example adapted from [19].

for the implementation of algebraic calculations that facilitate the classification of the topological features of the complex, and by extension, of the original topological space.

The algorithm begins with a data set of interest that consists of a point cloud X containing N points $x \in M$ sampled from a space M whose topology we want to elucidate. We define a real valued function f: $X \rightarrow R$ (f is referred in the literature as a *lens* or *filter*) whose value is known for the N data points. Next, we find the range I of the function f that is restricted to the points in X. We divide this range into a set S of smaller intervals of the same size, that overlap. This results in two

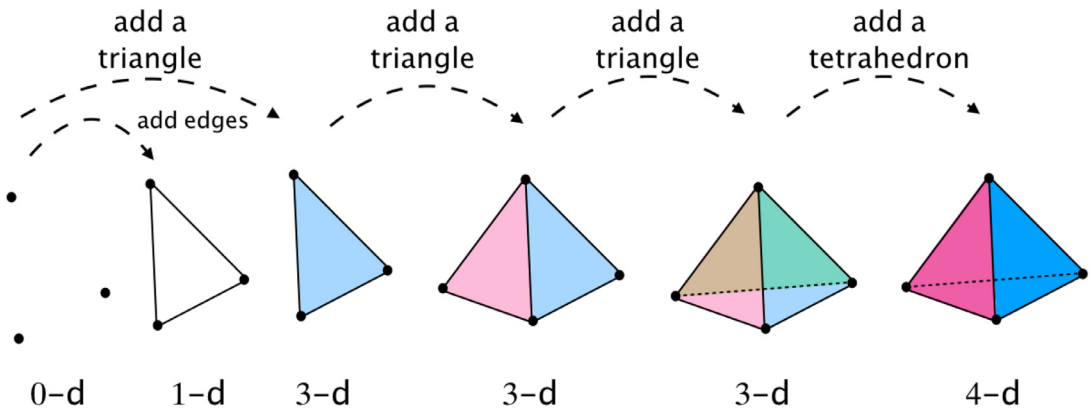


Fig. 3. A simple simplicial complex. A simplicial complex can be thought of as a generalisation of a network, that also includes triangles, tetrahedra and higher dimensional convex polyhedra.

parameters that can be used to control how detailed a representation of the data, i.e. the "resolution", namely the number l of the smaller intervals and the percentage overlap q between successive intervals.

For each interval $I_j \in S$, we find the set $X_j = \{x | f(x) \in I_j\}$, i.e. the points in X that form its domain. For each set X_j we form clusters $\{X_{jk}\}$, where $x_k \in X_j$ and $k >= 1$. We treat each cluster as a vertex in the resulting simplicial complex and draw an edge between vertices whenever $X_{jk} \cap X_{lm} \neq \emptyset$, for $X_{jk} \neq X_{lm}$ (i.e. when different clusters have non-empty intersection). Fig. 2 illustrates these steps for the construction of a topological network from data sampled from a 2-dimensional circle. The steps are implemented computationally using the Kepler mapper python library [25].

2.2. Simplicial complexes

To arrive at a precise definition of a simplicial complex and to understand it is not required for the context of this paper. Instead, here it is sufficient to think of simplicial complexes as a generalisation of networks that include higher dimensional elements. That is, simplicial complexes can be thought of as combinatorial objects consisting of vertices (0-d), edges (1-d), triangles (2-d), tetrahedra ("triangular pyramid") (3-d) and higher-dimensional (n-d for $n >= 0$) convex polyhedra. Fig. 3 illustrates this visually with a simple example. In practical terms, as described in [19], "using a single function as a filter we get as output a complex in which the highest dimension of simplices is 1 (edges in a graph). Qualitatively, the only information we get out of this is the number of components, the number of loops and knowledge about" local and global component structure. "One natural way of building higher dimensional complexes is to associate many functions with each data point instead of just one." In particular, "the dimension of simplices which we use to construct the complex" associated to two lenses "will be 4 or less" (depending on the covering of choice).

In practical terms regarding the mapper algorithm, when we use one lens to construct a simplicial complex, the maximum dimension of the simplices is 1, i.e. the complex is made up of vertices and edges and is thus a standard network. When we use two lenses to construct a simplicial complex, the maximum possible dimension of the simplices is 4, meaning it is made of vertices, edges, triangles and tetrahedra. Section 3.2 in the original paper of the mapper algorithm [19] describes explicitly how simplicial complexes are built with 2 lenses and how this can be generalised to more lenses and higher dimensional simplicial complexes.

2.3. Choosing the metric space(s)

The important aspect about the choice of metric for the purposes of the mapper algorithm is to have a notion of distance between two data points.

Although the mapper algorithm is less sensitive to the choice of metric than other methods that aim at creating simpler representative objects of spaces in high dimensions (e.g. dimensionality reduction algorithms) [19] we still wanted to make sure that our investigation was not biased by the choice of metric. Therefore we tested three metrics that "appeared suitable" for our data sets. These were the Euclidean, cosine and correlation metrics. We chose the Euclidean metric as it is the most intuitive to work with. Next, we wanted to investigate whether looping is a recurrent property in our data sets. In [15], where looping was indeed found to be a motif in their data set (of an infection from which patients can recover, such as in our study), the cosine metric was used, so we also decided to test this metric. Finally, we regarded the correlation metric to make the most biological sense. So we also tested this one.

Specifically, the distance is defined as follows for the three metrics we work with:

Euclidean: if $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ are two points in Euclidean n-space, then the Euclidean distance between them is given by the Pythagorean formula:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Computationally, this is done using the `sklearn.metrics.pairwise.euclidean_distances` function from Scikit-learn.

Cosine: if $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ are two points in Euclidean n-space, we use the following bespoke distance definition:

$$d(x, y) = \left| \frac{x.y}{||x||_2 * ||y||_2} - 1 \right|$$

Where $x.y$ is the dot product between x and y and $||u||_2$ is the L2-norm (Euclidean norm) of u. This definition is of the absolute value of the normalised dot product of x and y minus 1. The reason we take this definition and, not simply the normalised dot product, is the following: The normalised dot product of any two points takes values between $+1$ and -1. We want to cluster points whose vectors from the origin are almost parallel and point almost in the same direction. Two such points have a normalised dot product value that is close to $+1$. However, DBSCAN clusters two points whenever they are a distance less than or equal to the `eps` parameter. (The `eps` parameter is the maximum distance between two samples for them to be considered as in the same neighbourhood). Therefore, what we need is a distance formula that outputs small values for points that are "close" to each other (i.e. whose vectors from the origin point in roughly the same direction). The above bespoke distance formula achieves this. In order to implement this computationally, we precompute the distance matrix of the data using the normalised dot product with the `sklearn.metrics.pairwise.cosine_similarity` function of sklearn, next, we preprocess the distance matrix such that it fits our bespoke distance criterion:

```
X_cosine_similarity = sklearn.metrics.pairwise.cosine_similarity(X)
X_dist = np.abs(X_cosine_similarity - 1)
```

Finally, we pass the precomputed distance matrix to the clusterer, setting the `metric` parameter of DBSCAN to be equal to 'precomputed'.

Correlation: if $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ are two points in Euclidean n-space, then the correlation distance between them is given by the following formula:

$$d(x, y) = 1 - \frac{(x - \bar{x}).(y - \bar{y})}{||(x - \bar{x})||_2 ||(y - \bar{y})_2||}$$

Where $u.v$ is the dot product between u and v, $\bar{u}$ is the mean of the elements of u and $||u||_2$ is the L2-norm (the Euclidean norm) of u. Computationally, this is done using the `sklearn.metrics.pairwise_distances` function from Scikit-learn, setting the parameter `metric` to 'correlation'.

In the main part of this study we only reported the results obtained with the correlation metric. We observed the same results with the correlation and cosine metrics and we could not make concrete observations about the result obtained with the Euclidean metric. The questions of why the cosine and correlation metrics create the same simplicial complexes and how they differ from those created with the Euclidean metric require further investigations that were not done as part of this study.

2.4. Selection of lenses

The outcome of the mapper algorithm is highly dependent on the lens(es) chosen. For the purposes of our study, we categorised the lenses we used according to the information about the data they extract:

Features. These lenses are simply the values at each data point of a particular feature of interest. These were calculated using the `fit_transform` function of the Kepler mapper.

Distances to closest neighbours. These lenses report the distance of each data point to its n closest neighbours, or the sum of the distances to the n closest neighbours, under the metric of choice. These were calculated using the `sklearn.metrics.pairwise_distances` function from Scikit-learn, specifying the metric to be one of the three described above.

Dimensionality Reduction. These are projections of the data, usually, to the first (and possibly also the second) dimension(s) of various dimensionality reduction algorithms. These were calculated using the Manifold Learning algorithms from the Scikit-learn Python library and the `sklearn.decomposition.PCA` and `sklearn.decomposition.TruncatedSVD` functions of Scikit-learn. See Section 2.4.1 for a brief description of the methods used here.

Geometric properties. These report geometric properties. Specifically, we tested:

- The density using the `sklearn.neighbors.KernelDensity` function with Gaussian kernel and calculated the bandwidth using Scott's Rule [28].

– The eccentricity which is defined as follows:
Given p with $1 <= p < +\infty$, define

$$E_p(x) = \left(\frac{\sum_{y \in X} d(x,y)^p}{N} \right)^{\frac{1}{p}}$$

where $x, y \in X$. (Recall that we denote the data set of N points by X and $d(x, y)$ denotes the distance between $x, y \in X$, which is dependent on the metric of choice.)
– The infinite centrality which is a generalisation of the eccentricity above, for when $p = +\infty$, then $E_\infty(x) = max_{x' \in X} d(x, x')$.

Statistical properties. These report on statistical properties about the data points, such as the sum of the values of all the features for each data point, the average value of the features for each data point, etc. These were calculated using the `fit_transform` function of the Kepler mapper.

2.4.1. Dimensionality reduction algorithms

Here we provide short explanations about each of the dimensionality reduction algorithms used, (some of which have been adapted from the documentation of Scikit-learn [29]).

Singular value decomposition (SVD) and Principal Component Analysis (PCA) are two methods used to perform linear dimensionality reduction of high-dimensional data set.

PCA reduces the data into linearly uncorrelated variables (called *principal components*) such that the first component accounts for as much of the variability in the data as possible, and each succeeding component in turn accounts for the next highest variance possible and is orthogonal to the preceding components. When used in dimensionality reduction, one can take the first few principal components as the new set of features of the data set. To implement this, we used `sklearn.decomposition.PCA`.

SVD is a matrix decomposition method for reducing a matrix A to the product of its constituent parts ($A = U \cdot \Sigma \cdot V^T$, where U, V are unitary matrices and Σ is a rectangular diagonal matrix of singular values). When applied in dimensionality reduction, one can select the top k largest singular values in Σ and use the k affiliated columns in Σ and rows in V^T to generate an approximation $B = U \cdot \Sigma_k \cdot V_k^T$ to A. We used the TruncatedSVD class of Scikit-learn to implement this.

LLE, Isomap, MDS and Spectral Embedding are algorithms that belong to a framework called Manifold Learning, which are an attempt to generalize linear frameworks like PCA, to be sensitive to non-linear structure in data. These approaches learn the high-dimensional structure of the data from the data itself, in an unsupervised manner, i.e. without the use of predetermined classifications.

The Isomap seeks a lower-dimensional embedding which maintains geodesic distances between all points. LLE seeks a lower-dimensional projection of the data which preserves distances within local neighborhoods, and it can be thought of as a series of local Principal Component Analyses which are globally compared to find the best non-linear embedding. For the Isomap and the LLE one needs to make a choice on the number of neighbors; since in our data set each time point should have between two and five data points corresponding to individually analyzed mice, we chose the value of three neighbors for these algorithms.

Multidimensional scaling (MDS) seeks a low-dimensional representation of the data in which either the distances between the two output points are set to be as close as possible to the similarity or dissimilarity of the original data (the metric approach) or the algorithm seeks a monotonic relationship between the distances in the embedded space and the similarities/dissimilarities of the original data (the non-metric approach).

For spectral embedding Laplacian Eigenmaps are implemented, which find a low dimensional representation of the data using a spectral decomposition of the graph Laplacian. The graph generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. Points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances.

2.5. On the number of intervals and percentage overlap for each lens

The choice of number of intervals and percentage overlap defines how detailed or coarse a topological network representation of the data we want to create is, in other words, the "resolution" of the representation. The number of intervals can be any number starting from 1 and, as the name indicates, the percentage overlap can be anything in the range [0.0,1.0]. Thus, choosing a higher number of intervals translates to increasing the number of vertices of the graph. Increasing the percentage overlap allows the algorithm to find more data points in common between two intervals, thus roughly translating to an increased possibility of connecting two vertices. (Note that increasing the number of lenses also increases the number of vertices, since it increases the number of intervals that the data is partitioned into). Fig. 4 shows topological networks of data sampled from a 2-dimensional circle, at different resolutions.

2.5.1. Scaling of the data

PCA gives different results when the scales of the features are different. Additionally, manifold learning methods are based on a nearest-neighbor search, therefore, such algorithm may perform poorly if the features of the data are on different

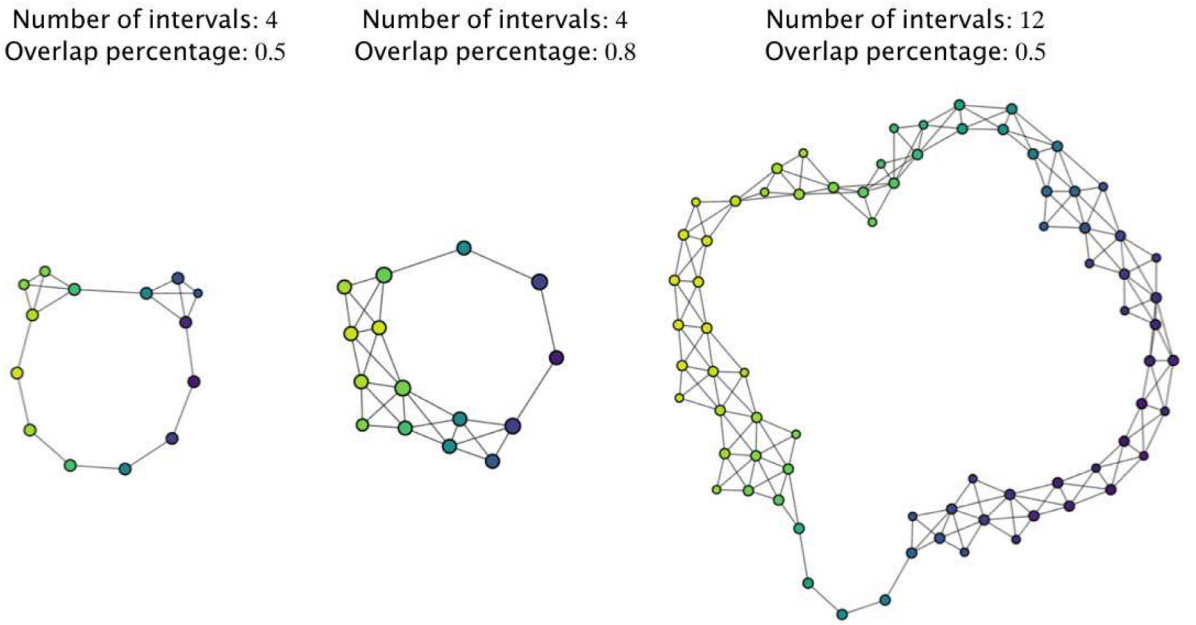


Fig. 4. Resolution of simplicial complexes and persistence of the shape of the data. Three simplicial complexes of data sampled from an unknown 3-dimensional space of different resolution are generated using the mapper algorithm [19], implemented computationally using the Kepler mapper Python library [25]. One lens is used for all simplicial complexes, namely a projection on the x-axis. The number of intervals and percentage overlap for the lens at each resolution are indicated. Observation 1: the three simplicial complexes consist of one connected component. Observation 2: the three simplicial complexes have a "big hole" in the middle. Conclusion: The data set is sampled from a topological space that consists of one component and that has a hole in the middle. Since we know it is a 3-dimensional space, we can say it is a torus.

scales. Therefore, we normalise the data before applying the dimensionality reduction algorithms. (Note that the clustering algorithm is run on the original data, which we do not normalise.)

2.5.2. Choice of clustering algorithm

Finding a good clustering of the points is a fundamental issue in computing representative topological networks. Currently there is no automated or principled method of making a choice. The mapper algorithm does not have any limitations on the cluster used and in particular, according to [19] desired characteristics when choosing a clustering are that it:

1. is able to take an interpoint distance matrix as an input, not restricted to the Euclidean distance,
2. does not require specifying the number of clusters.

We chose the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [29] because we can specify what metric to use, the thresholds for the number of points needed to create a cluster (`min_samples`) and the value of the distance below which two points are considered to belong to the same cluster (`eps`).

2.6. Grid search analysis of parameter values for the mapper algorithm

As described in the previous section, the outcome of the mapper algorithm is highly dependent on the choice of values for the following parameters: the number of lenses, the type of lenses, the number of intervals for each lens, the percentage overlap for the intervals, the choice of metric space and the clustering algorithm.

We wrote a semi-unsupervised algorithm that built all simplicial complexes for various metrics, lenses and ranges of values for the lens intervals, percentage overlap and epsilon values of the DBSCAN clusterer from sklearn, and which "chose appropriate" simplicial complexes to represent the data. More specifically, below we list all the values we tested.

Metrics: cosine, euclidean and correlation

Number of lenses: 2

To start with, we constructed simplicial complexes of the data sets with between 1 and 4 lenses. However, when we visually inspected the result of using a three or four lenses, we observed many more clusters corresponding to the same points, because of oversampling by the larger number of intervals introduced by the third and fourth lenses. In addition, using more that two lenses can make the calculations take over 20 hours to complete, compared to 3-5 hours for two lenses (ran on a standard laptop). Therefore we limited ourselves to using two lenses.

Combinations of lenses:

1. Lens 1 = Distance to the first neighbour with Lens 2 = Distance to the second neighbour
2. Lens 1 = Distance to the first neighbour with Lens 2 = Projections to features
3. Lens 1 = Sum of the distances to the first and second neighbours with Lens 2 = Projections to features
4. Lens 1 = First dimension of a dimensionality reduction algorithm with Lens 2 = Projections to features
5. Lens 1 = First dimension of a dimensionality reduction algorithm with Lens 2 = Second dimension of the same dimensionality reduction algorithm
6. Lens 1 = Geometric or statistical information with Lens2 = Projections to features

The projections to features lenses are a projection to each of the nine features of the data sets (the two pathogen loads and the seven cytokine concentrations). The distance to the closest neighbours and the geometric and statistical lenses depend on the metric chosen.

Number of intervals: between 2 and 30.

Percentage overlap: between 0.1 and 0.9.

Clusterer: DBSCAN from scikit-learn [29]. For the `min_samples` parameter we chosed the value of 1, i.e. a cluster can be formed with 1 or more data points. For the `eps` parameter we tested values between the minimum and maximum distances in the data sets for each metric. More specifically, the distances between the data points in the three infection groups range between [0,520002] in the Euclidean metric; between [0,1] in the cosine metric; and between [0,2] in the correlation metric. So for the Euclidean metric we tested 5000 values between 0 and 520002; and for the cosine and correlation metrics we tested 10 values between 0 and 1 and 0 and 2, respectively.

Taking all these together, there are 18 different pairs of lenses; 28 different intervals and 10 percentage overlap values to test for each lens; 10 values for the epsilon parameter for the clusterer for the cosine and correlation metric and 5000 for the Euclidean metric. Therefore, in total this parameter grid search exercise generated almost 1 billion simplicial complexes ($15 * 28^2 * 9^2 * 10 + 3 * 28^2 * 10^2 * 5000 = 962,085,600$).

From all the simplicial complexes generated, the algorithm chooses those that have a user-specified number of connected components. Biologically it makes sense that the simplicial complexes for the three infection groups have only one connected component. From the resulting list of simplicial complexes with one connected component, we ordered the simplexes in ascending order of epsilon value for the clusterer and the percentage overlap for the intervals for both lenses; choosing the smaller rather than the larger values for these parameters makes the simplicial complexes, as models of the system, simpler, which is desirable when working with models. Next, we chose simplicial complexes that had number of vertices at least half of the number of data points in the data set (for example, if the data set for the IAV infection group has 30 data points then we favour simplicial complexes that consist of only one connected component and at least 15 vertices); choosing this number gives a good resolution, not too simple, but also prevents the user from choosing simplicial complexes that have oversampling of data points. Finally, we visually inspected the simplest (in terms of small values for the parameters) 10, 20 or 30 simplicial complexes in that list. The visual inspection has two purposes: First, to give the user an idea of whether there is persistence in the global and local structures of the simplicial complexes generated. Second, to choose a representative structure for the data set.

2.7. Co-infection experimental data

We consider the murine data that we first presented in [4]. Three groups were considered: single viral infection (IAV), single bacterial infection (T4) and co-infection (IAV + T4). All experiments were performed in groups of 4–7 WT C57BL/6 J mice. The mice were anesthetized and intranasally infected with either a sublethal dose of IAV (A/PR8/34) or a bacterial infection with the *S. pneumoniae* strain T4 on day 7, or both, depending on the infection group. Following infection, mice were monitored daily for morbidity and mortality. Bronchoalveolar lavage (BAL), post-lavage lung and blood were collected at 1.5, 6, 18, 26 and 31 hours post bacterial infection (hpi) or post bacterial co-infection (hpc). Lungs were homogenized, and the supernatants were used to determine virus titers, immune cell populations, and cytokine and chemokine concentrations. Kinetic measurements for viral titers (mRNA by real-time PCR), bacterial counts (colony forming units (CFU)) as well as the following cytokines were considered: IFN-γ, TNF-α, IL-6, IFN-β, IL-22 and the chemokines MCP-1 and GM-CSF. The experiments are described in detail in [4].

For the analysis done in this study we took the measurements collected from the post-lavage lung. For the T4 and AIV + T4 infection groups, at 18 hpi and 18 hpc, respectively, the measurements had to be repeated. As it is a cross-sectional study (each measurement is coming from a mouse), to impute corresponding values, we allocate high values of bacterial burden to rows that contain high values of cytokines. Since mice were naive, we replaced any N/A values (under level of detection) in the bacterial burden (in the single IAV infection group) and viral load (in the single T4 infection group) with the value 0. For each infection group, for time points that have less than five missing values in one feature, we replaced the missing values with the average value of the feature for that specific time point. For infection groups that have missing values at one single time point for one particular feature, we performed a linear interpolation between the mean values of the previous and the next time points and replaced the missing values with the predicted value for that time point.

Nodes in the simplicial complex represent clusters of infected mice, and edges connect nodes that contain samples in common. Nodes are colored by the average value of their samples for the variables listed in the Figs' legends and color maps.

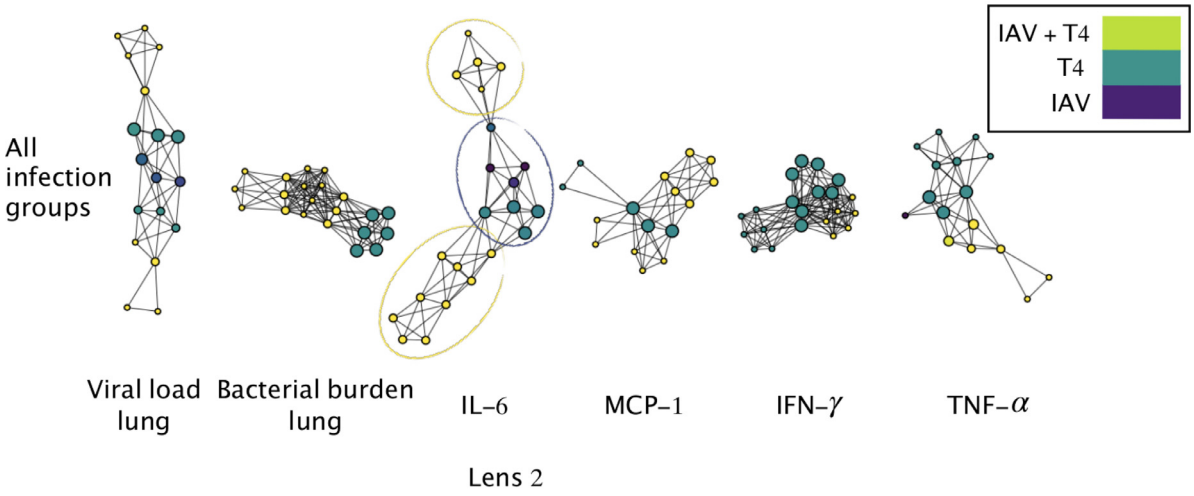


Fig. 5. Simplicial complexes of data set consisting of all infection groups. The vertices of the simplicial complexes are color coded according to the infection group of the data points in the clusters. The legend shows which colors correspond to which infection group. Clusters that contain data points belonging to more than one infection group are colored by the average color value for members in that node.

Three types of parameters are needed to generate a topological model: First is a notion of similarity, called a metric, which measures the distance between two points in some space (in our case, the points are the rows in the data and the space is a multidimensional space that can be plotted using the quantitative measurements of disease symptoms as axes, such as the pathogen load or cytokine concentration, i.e. the features of the data set). The metric we used is the correlation distance. Second are lenses, which are functions that describe the distribution of data in a space. A lens is a mathematical mapping (function) that converts a data set into a vector, in which each row in the original data set contributes to a real number in the vector; i.e. a lens operation turns every row into a single number. Metrics are used with lenses to construct the simplicial complex output. Multiple lenses can be used in each analysis. In this case, Keppler-Mapper handles them mathematically by considering the Cartesian product. Third is the resolution, which controls the number of bin partitions that will be created within the range of selected lens values, known as the number of intervals, and the amount of oversampling between bins, known as the percentage overlap. Clustering then takes place within the bins, forming the final vertices of the simplicial complex; and clusters are connected with an edge whenever they share data points within the region that are over-sampled according to the percentage overlap. Therefore increasing the number of bins increases the number of vertices and increasing the percentage overlap results in an increased number of edges. The metric, lenses, resolution, and clusterer used to generate the topological graphs in Figs. 5 and 6 are as indicated in Table 1. K-Nearest Neighbour Analysis was implemented using the KNeighborsClassifier from the Scikit Learn Python library [29], with the correlation metric. T-tests were done using the ttest_ind function from the SciPy Python library.

Table 1 shows the parameter values used with the Kepler mapper to generate the simplicial complexes of Fig. 5 and 6 in the main script.

3. Results

In the context of infectious diseases from which hosts can recover (e.g. malaria, influenza, pneumococci, etc), disease maps that form loops are representative of the trajectory from health, through infection and sickness and back to recovery or death [15]. In particular, disease space with looping behaviour and disease maps that are circular can be used to, for example: 1.) describe the in-host dynamics of infections; 2.) identify where data of patients lie along the infection timeline, regardless of whether the stage of the infection is known (for example, from cross-sectional studies or from data of patients coming clinics to be treated, at different (and unknown) stages of the infection); and 3.) distinguish between more and less resilient individuals, from a relatively early stage of the infection course. This methodology was applied in [15] to investigate malaria. Motivated from this approach, we investigated if looping curves are a common motif in influenza or influenza in co-infection with bacteria. To that end, we also generated phase plots of pairs of features of our data sets and used K-nearest neighbour analysis, as is explained below.

Definition 3.1: *Disease space* is the multidimensional space that can be plotted using quantitative measurements of disease symptoms as axes.

Definition 3.2: Parameters of a data set that oscillate, partially overlap and have a time lag between them are called *hysterectic parameters* [15].

Definition 3.3: When pairs of hysterectic parameters are plotted against each other in their phase plot they create loops [15]. These can be considered two dimensional projections of the higher dimensional looping behaviour of the data set and are referred to as *disease maps*.

Table 1
Parameter values for the Kepler mapper to generate the simplicial complexes of Fig. 5 and 6 in the main script. The metric used was correlation.

Lens 1	Lens 2	Num. intervals Lens 1	Num. intervals Lens	Percentage overlap Lens 1	Percentage overlap Lens 2	Clusterer eps
All infection groups						
Dist. 1st Neighbour	Projection viral load lung	1	13	0.3	0.8	0.5
Dist. 1st Neighbour	Projection bacterial burden lung	3	7	0.8	0.7	0.5
Dist. 1st Neighbour	Projection IFN-γ	5	3	0.8	0.8	0.5
Dist. 1st Neighbour	Projection TNF-α	1	9	0.2	0.8	0.5
Dist. 1st Neighbour	Projection MCP-1	5	3	0.8	0.4	0.5
Dist. 1st Neighbour	Projection IL-6	1	13	0.3	0.8	0.5
IAV						
Dist. 1st neighbour	Projection viral load lung	1	13	0.2	0.8	0.5
Dist. 1st neighbour	Projection IFN-γ	3	3	0.6	0.8	0.5
Dist. 1st neighbour	Projection TNF-α	3	3	0.8	0.7	0.5
Dist. 1st neighbour	Projection MCP-1	1	9	0.2	0.8	0.5
Dist. 1st neighbour	Projection IL-6	3	3	0.6	0.8	0.5
T4						
Dist. 1st neighbour	Projection bacterial burden lung	3	3	0.7	0.8	1.1
Dist. 1st neighbour	Projection IFN-γ	3	3	0.7	0.8	1.1
Dist. 1st neighbour	Projection TNF-α	1	13	0.2	0.8	1.1
Dist. 1st neighbour	Projection MCP-1	3	3	0.7	0.8	1.1
Dist. 1st neighbour	Projection IL-6	3	3	0.7	0.8	1.1
IAV + T4						
Dist. 1st neighbour	Projection viral load lung	1	13	0.3	0.8	0.5
Dist. 1st neighbour	Projection bacterial burden lung	3	7	0.7	0.7	0.5
Dist. 1st neighbour	Projection IFN-γ	3	3	0.8	0.7	0.5
Dist. 1st neighbour	Projection TNF-α	5	3	0.8	0.6	0.5
Dist. 1st neighbour	Projection MCP-1	3	3	0.8	0.4	0.5
Dist. 1st neighbour	Projection IL-6	3	3	0.8	0.5	0.7

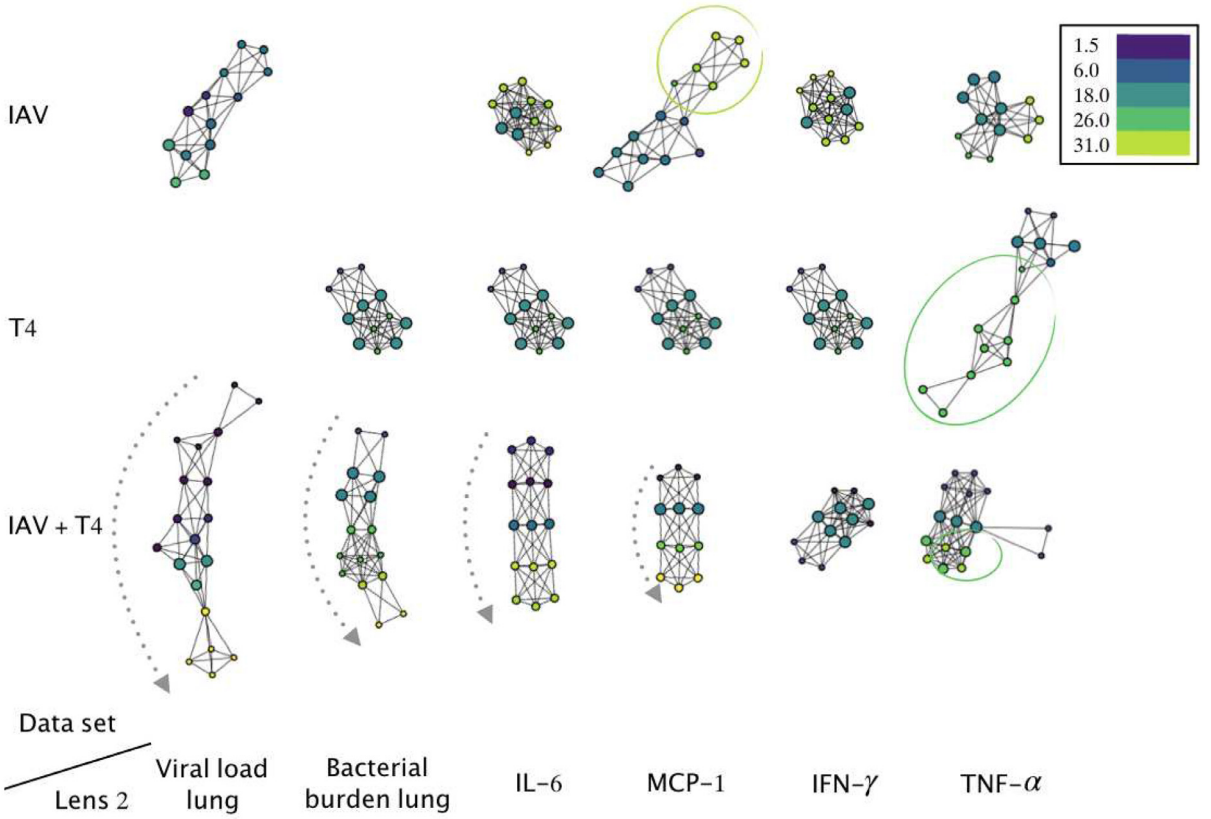


Fig. 6. Decoupling simplicial complexes of the immune response to virus, bacteria and co-infection. Five simplicial complexes are generated for the single viral (top row) and single bacterial (middle row) infection groups and six simplicial complexes are generated for the co-infection group (bottom row). The vertices of the simplicial complexes are color (see legends) coded according to the hour post infection or co-infection in the clusters. Clusters that contain data subsets belonging to more than one time point are colored by the average color value for members in that node.

3.1. Persistent shape of the data in the three infection groups

We used the mapper algorithm to study four data sets: The first consisted of the data for the three infection groups put together, and the other three were the individual separate infection scenarios. The Kepler Mapper [25], a library implementing the mapper algorithm in Python, was employed. Additionally, we wrote a semi-unsupervised algorithm that built all simplicial complexes for various metrics, lenses and ranges of values for the lens intervals and percentage overlap, and which chose simplicial complexes to represent the data. More specifically, using our semi-unsupervised algorithm, we tested the cosine, euclidean and correlation metrics, along with different epsilon values for the clusterer. Furthermore, all analysis was done using two lenses and we tested different combinations of the following: projections to the features of the data sets (i.e. the values of the pathogen load or the concentrations of the cytokines), the distance to the two nearest neighbours, the first two dimensions of various linear and non-linear dimensionality reduction algorithms and projections to (the image of) functions that reveal interesting geometric and statistical information about the data, such as density, eccentricity or centrality. We tested between 2 and 30 intervals for each lens and 10 different values for the percentage overlap of the lens' intervals and the epsilon parameter of the clusterer. This resulted in almost 1 billion simplicial complexes being generated, from which, using properties of graphs such as the number of connected components, the algorithm chose complexes that were persistent in shape and that showcased important information about the immune response.

A detailed presentation of our semi-unsupervised algorithm and a discussion on how the representative simplicial complexes are chosen and how we came to the conclusion that those (simplicial complexes) are persistent in their shape is included in Section 2. Of note, during the parameter value search we obtained consistent results to those presented here with the same lenses and metric (correlation) and different parameter values for the number of intervals and percentage overlap. We also obtained similar results with the same metric but other pairs of lenses that included also linear and non-linear dimensionality reduction algorithms. Finally, we obtained similar results also with the cosine metric.

For the sake of clarity, we discuss only the results obtained by using the correlation metric with the following two types of lenses: lens 1 is the distance to the first neighbour and lens 2 is a projection to one of the features. Figs. 5 and 6 illustrate the representative simplicial complexes that we discuss in more detail here (Table 1 lists the parameter values specific for

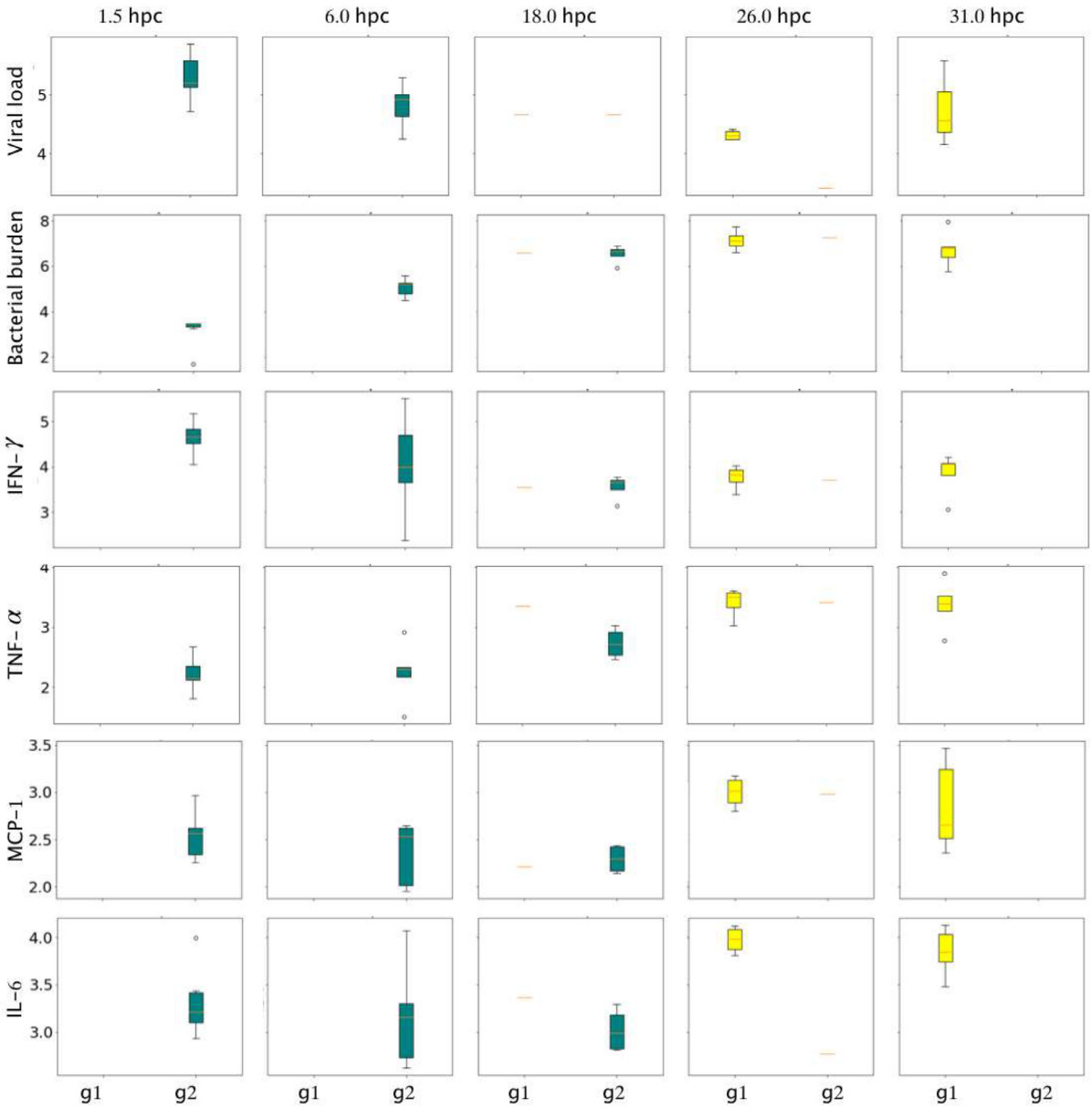


Fig. 7. Box plots of data points that belong to the two distinct regions revealed by TDA. $g1$ denotes the data points belonging to vertices in the yellow group in Fig. 5. $g2$ denotes the data points belonging to vertices in the teal/purple group in Fig. 5.

these simplicial complexes). Fig. 5 corresponds to the data set that consists of all infection groups together, and Fig. 6 shows the simplicial complexes for the individual infection groups separately. The columns in both figures indicate the projection for lens 2.

These analyses revealed persistence in the shape of the data. For example, in Fig. 5 all the simplicial complexes generated with the different projections for lens 2 can be divided into three regions, two in yellow and one in purple/teal, where the circles (or vertices) in the yellow regions belong to time points 26 and 31 h post co-infection (hpc) in the IAV + T4 infection group, and the circles in purple/teal belong to the IAV, T4 single-infection groups and early (1.5, 6, 18 hpc) time points in the IAV + T4 co-infection group. This is illustrated more explicitly in the simplicial complex generated by lens 2 = IL-6. The fact that the same shape is generated regardless of the projection used indicates that the shape is likely to represent the data.

Similar observations can be made for the simplicial complexes of the individual infection groups in Fig. 6. For example, the simplicial complexes of the IAV infection group (second row) generated with the cytokines can all be divided into two regions, green vertices versus blue vertices. The simplicial complexes of the single bacterial infection (T4 infection group)

Table 2
Statistical study to data sets belonging to consecutive time points (columns) for each feature (row). p-values below 0.05 (*), 0.01 (**) and 0.005 (***) are indicated.

Infection group	Feature	1.5–6	6–18	18–26	26–31
IAV + T4	Viral load lung	0.113336	0.202011	0.000101***	0.200436
	Bacterial burden lung	0.009209**	0.006859**	0.104765	0.958299
	IFN-γ	0.748814	0.269034	0.161643	0.320453
	TNF-α	0.520351	0.120874	0.016524*	0.766774
	MCP-1	0.411175	0.353125	0.000691***	0.859898
	IL-6	0.879291	0.425412	0.019389*	0.994039

generated with the bacterial burden and the cytokines IL-6, MCP-1 and IFN-γ are also all persistent in shape and, unlike the simplicial complexes for the other data sets, these ones do not highlight particular groups of nodes. The bottom row of Fig. 6 corresponds to the simplicial complexes of the co-infection group (IAV + T4) alone. The time course can be clearly distinguished by the simplicial complexes generated with lens 2 being the two pathogens burden and the concentration of the cytokines IL-6 and MCP-1, as is indicated by the gray arrow (from early to later time points).

The simplicial complexes reveal different persistent shapes for the three infection groups - in IAV the later time points are segregated from the earlier time points; in T4 the simplicial complexes are homogeneous and do not reveal special areas; in IAV + T4 the time course of the infection is elucidated. Together this indicates that the immune system behaves differently in the three infection scenarios.

3.2. Transition points in the immune response

We originally applied the mapper algorithm to all infection groups together in order to see whether the three infection groups would be clearly separated. However, the result illustrated in Fig. 5 contains far more information. Taking the complex generated with the projection to IL-6 as the representative shape of the data, we observe that it has three regions: two in yellow and one in purple/teal.

The data points that have been clustered (by the Mapper Algorithm) into the yellow vertices of the simplicial complex correspond exclusively to all the late (26 and 31 hpc) time points of the co-infection group (IAV + T4); the rest of the data points (all other time points in the co-infection and all the data in the single viral (IAV) and single bacterial (T4) infection groups) have been grouped together by the mapper algorithm and clustered into the vertices colored teal/purple. In other words, the Mapper Algorithm has segregated those specific points (26hpc and 31hpc in the co-infection) away from all other points, indicating that the data behaves differently at and after 26hpc in the co-infection scenario, compared to the other points. This, we interpret as the immune system undergoing a shift in its behaviour in the co-infection, sometime between 18 and 26 h post co-infection. From this observation we conclude that the topological data analysis has highlighted a transition in the nature of the immune response during a co-infection sometime between 18 and 26 hours post co-infection.

In order to interpret more precisely why the mapper algorithm is highlighting the specific regions illustrated inside the yellow circles, we calculated the p-values between groups of data points at consecutive time points, for each feature of the data (Table 2); we also made box plots to compare the data belonging to the two groups separated by the simplicial complex in Fig. 5 corresponding to lens 2 = IL-6 (Fig. 7). The p-values show that, in the co-infection scenario, there is a strong change in the concentration of the cytokines TNF-α, MCP-1 and IL-6 (as well as in the viral load) between 18 and 26 hpc. The boxplots clearly revealed that data points of IAV + T4 from 26 and 31 h post co-infection are separated by the simplicial complex from the earlier data points and that 18 and 26 hpc represent a transition point for the system, since at both time points there are data subsets that belong to both groups in the simplicial complex (the yellow and the teal/purple). It is possible that the shape of the simplicial complexes in Fig. 5 is exactly mirrowing the dramatic change in the concentration of the cytokines (and possibly the viral load) that is revealed by the *p*-values.

Fig. 8A shows the k-nearest neighbours for the data sets in the three infection groups, with metric correlation and 30 neighbours. The rows and columns indicate each data points and dark regions indicate the 30 closest neighbours of each data point. The labels show which points belong to which infection group and time point. The bottom right panel, inside the orange dotted box, corresponds specifically to the data points of the co-infection group (IAV + T4). The data sets in this region can be divided into three groups: the early period (1.5 hpc), the transition period (6 hpc) and the later period (between 18 and 31 hpc). This is illustrated by the fact that within the orange dotted box, the data sets at 1.5 hpc are neighbours with only data sets at 1.5 and 6.0 hpc, the data sets in 6 hcp have neighbours in both the early and the later groups and data sets at 18, 26 and 31 hpc have neighbours only at 6 hpc and late time points. This can be seen more clearly in Fig. 8B which shows the correlation coefficients between data points in the co-infection data set. Data sets at 1.5 hpc are closely correlated to themselves, data sets at 6 hpc have close correlation to points in both the early and later periods, the later data sets (18, 26 and 31 hpc) are closely correlated to themselves only. Data sets between the early and later groups are not correlated.

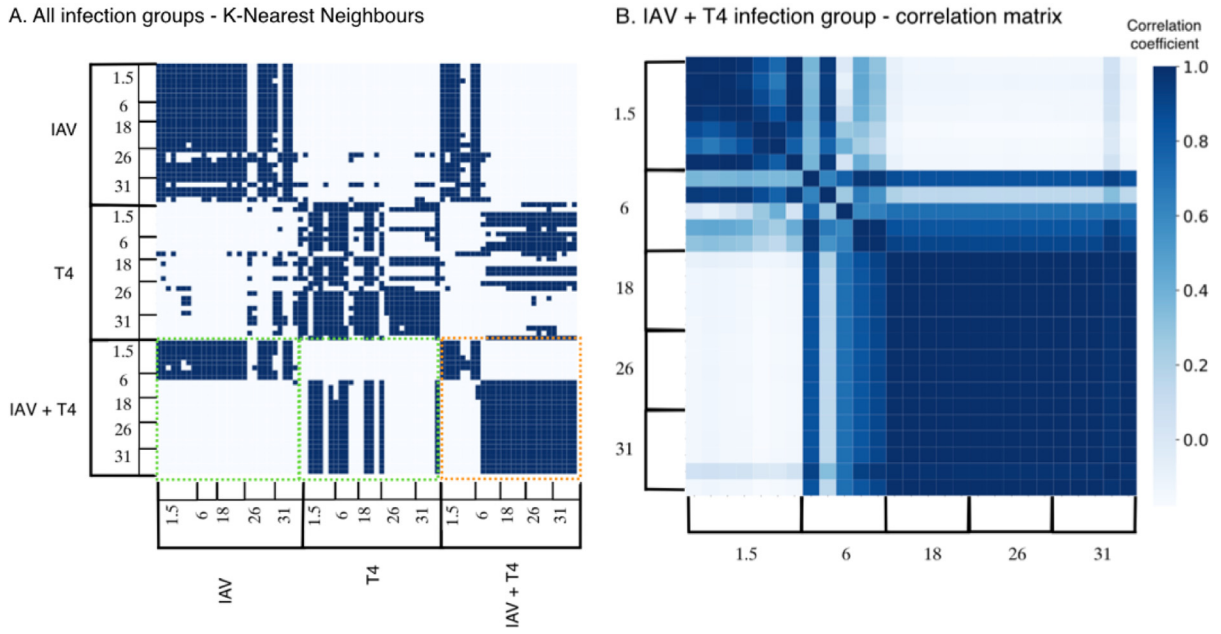


Fig. 8. K-nearest neighbour analysis of co-infection. A) 30-nearest neighbours of all data sets in the three infection scenarios, with the correlation as metric. The infection groups and the time points for each infection group are indicated on the axes. The dark regions indicate the 30 neighbours of each data point. Inside the orange dotted box are the neighbours of co-infection data within the co-infection scenario. Inside the green dotted boxes are the neighbours of the co-infection data in the IAV single infection and T4 single infection groups. B) Correlation distance matrix of the co-infection group. The color-bar on the right indicates the correlation distance values between data sets.

In Fig. 8A, now looking at the neighbours of the co-infection data subsets outside of the orange dotted box and inside the green dotted boxes (i.e. along the rows), we can see that co-infection data sets at early times also have neighbours exclusively in the IAV single infection group and co-infection data sets at late times also have neighbours exclusively in the T4 single infection group. In other words, the early co-infection data sets are closely correlated with the data sets in the IAV single infection group and the late co-infection data sets are closely correlated with the data sets in the T4 single infection group. Putting these three observations together, we can conclude that the immune response in the co-infection inherits its early response from the primary viral infection but at 6 hpc it undergoes a shift and quickly shapes its response to defend against the secondary bacterial infection.

In the simplicial complexes illustrated in Fig. 6 in both the T4 single infection and co-infection (IAV + T4) groups the simplicial complexes generated with the TNF-α projection draw attention to the later time points in both infection scenarios. Specifically, for T4, the points correspond to time 26 hpc and in IAV + T4 the data sets correspond to times 26 and 31 hpc. Table 2 highlights that in both the single bacterial infection and in the co-infection, the concentration of the TNF-α cytokine changes significantly between 18 and 26 hours post the onset of the bacterial infection. Our previous analysis [4] of experimental results for TNF-α showed levels were increasingly and significantly elevated in co-infected mice from 18 hpi on. Therefore the simplicial complexes in Fig. 6 of the single bacterial infection and the co-infection generated by the projection to TNF-α are exactly revealing this dramatic change in the concentration of this cytokine at this late stage in the course of the co-infection.

4. Discussion

The relative contributions of the immune system during co-infections and how they can help in laying out the evolution of the immune system in response to co-infections are largely fragmented. The complexity of multi-pathogen infections makes detailed dissection of contributing mechanisms and stages of the immune response, which may be non-linear and occur on different time scales, challenging. Recently, in conjunction with experimental data, theoretical approaches have been able to uncover infection control mechanisms, establish regulatory feedback, connect mechanisms across time scales, and determine the processes that dictate different disease outcomes [30]. In this study we aimed at continuing this effort and we used TDA and data of co-infection experiments [4] to investigate how the immune system evolves between different infections.

Using the Mapper Algorithm (Figs. 5 and 6) in combination with nearest neighbour analysis (Fig. 8) we have shown that the immune response during influenza-pneumococcal co-infection consists of three stages (Fig. 1): It is initially shaped by the inherited response to the primary influenza infection. We call this phase 1 of the immune response. Subsequently, the

system undergoes an abrupt transition at 6 hours post onset of the secondary bacterial infection as it quickly modulates itself and starts responding predominantly to the bacterial infection; this is phase 2 of the immune response. There is a second transition stage between 18 and 26 h post co-infection after which the immune response does no longer resemble the behaviour under a single viral or bacterial infection, but presumably shapes its response to the co-infection itself; this stage we call phase 3.

In [4], kinetics of bacterial growth and clearance in the respiratory tract and blood following IAV-*S. pneumoniae* co-infection revealed a turning-point between 6 and 18 h post onset of the secondary bacterial infection. In this study we have narrowed down the time of the turning point specifically to 6 hours post co-infection.

Experimental results [4] for IFN-γ showed that the co-infection led to an increase as early as 1.5 hpc and 6 hpc compared to the single IAV infection. The levels of IFN-γ remained constant compared to the underlying IAV infection for the later time points, but a significant increase was observed when compared to the single T4 infection. Finally, overshooting concentrations of IL-6 in the co-infected mice were also detected experimentally at 26 hpc and 31 hpc compared to the single T4 infection. The chemokine MCP-1 was experimentally found to be significantly increased in the IAV + T4 group compared to the single T4 infected group and marginally increased to the IAV only group at 26 hpi and 31 hpi.

The simplicial complexes illustrated in Fig. 6 perfectly matches these observations. More specifically, the experimental observations made in [4] regarding the temporal changes in concentrations of the cytokines throughout the co-infection coincide with the special regions highlighted by the simplicial complexes. We observe that the simplicial complexes generated by the mapper algorithm with a projection to the features could be representing exactly those dramatic changes in the values of the features and that the algorithm is able to separate data points with high relative concentrations of cytokines away from other data points. In the simplicial complexes illustrated in Fig. 6 in both the T4 and IAV + T4 infection groups the simplicial complexes generated with the TNF-α projection draw attention to the later time points in both infection scenarios. Specifically, for T4, the data sets correspond to time 26 and in IAV + T4 the data correspond to times 26 and 31. We could further interpret these results as further supporting evidence that the immune response at this stage of the co-infection is primarily responding in a way that is similar to its response in the single bacterial infection.

The simplicial complex of the co-infection generated with the projection to IFN-γ in Fig. 6 segregates data points at early times 1.5 and 6 hpc (in purple) away from the clusters of the other data sets, resembling the experimental observations regarding the concentration of IFN-γ in the co-infection compared with the single viral and bacterial infections. The simplicial complexes of the co-infection scenario generated by the projections to IL-6 and MCP-1 (and to the viral load and bacterial burden) reproduced the timeline of the infection course. It is interesting to contemplate the possibility that the simplicial complexes of the co-infection imply that the cytokines IL-6 and MCP-1 play a consistent role through the whole infection course in the co-infection scenario.

Looking in more detail at the simplicial complex corresponding to the projection to IL-6 on the top row of Fig. 5, we can see that the two groups of yellow circles are sprouting from different regions in the complex; one yellow group sprouts from vertices in purple and the other from vertices in teal. Recall that the vertices of the simplicial complex represent clusters of points of the data set and edges between vertices indicate that there are data in common between clusters. Therefore, after further analysis of the data sets that are in common between the purple and yellow vertices (clusters) and teal and yellow vertices, we could elucidate that, in fact, the purple vertices have data sets exclusively in the single viral infection at early time points and vertices in teal have all the data sets in the single bacterial infection and some data sets in the late stages of the single viral infection. In other words, the yellow region emanating from the purple vertices is connected exclusively to early time points in the single viral infection and the other yellow region to the single bacterial infection and late time points in the viral infection.

We may speculate the connections between the late-stage co-infection data and the early-stage single viral infection data are hinting at a rebound in viral titre after bacterial infection is established, which is a property of the co-infection that was been observed in other studies (for example) [31].

In [4] mathematical modelling, we proposed IFN-γ as a key and sufficient modulator in the impairment of bacterial clearance and other detrimental effects specifically for IL-6 and TNF-α in bacterial clearance. At the current stage of application of the mapper algorithm for this particular data set, we are not able to draw specific conclusions regarding causal roles of the cytokines in the co-infection. We can only allude to possible involvement of cytokines at specific stages in the infection course (as we have done, for example, with TNF-α, IL-6 and MCP-1.) In other words, TDA can dissect the potential of the different cytokines to represent the whole data set during co-infections, however, it can not point out or reject a key role of a specific cytokine for the susceptibility to bacterial co-infections.

The simplicial complexes generated for T4 single infection with projections to all the features show a homogenous structure where no specific group of data points are segregated or highlighted. We believe this is due to a trichotomy of pneumococcal outcomes discovered using stability and bifurcation analysis in [23]. Additionally, the immune response has been found to go through three stages during its response to single pneumococcal lung infections [32].

In [15] the simplicial complex of the stages a host infected with malaria goes through is circular and serves as a map of the loop an individual goes through on its way from health, through sickness and recovery and back to health. It is to be expected that a topological approach to study infectious diseases where hosts recover would also reveal circular topological simplicial complexes. This is not the case for the data set we have used. For the three infection groups, the data sets do not contain information for the full course of the infections. More specifically, the data set of the single IAV infected group is based on data starting from day 7 post the onset of the viral infection. The data set for the co-infected groups is incomplete

towards the end of the infection course because for ethical reasons the mice that developed a high morbidity had to be euthanized before the bacterial infection is resolved naturally. Nevertheless, we found hints of looping behaviour in the co-infection, as discussed in Section 3.1.

In some cases, it has been difficult to uncover the implications in the biological context with certainty, with respect to the structures of the simplicial complexes. For example, we speculate that the segregation of specific data sets represents striking changes in feature values - i.e. changes in concentration of cytokine of pathogen load from one time point to the others. This seems indeed to be the case as explained earlier in this discussion. However, for example, striking changes in bacterial burden and viral load in the co-infection are shown in Table 2, where the bacterial burden changes significantly between 1.5 and 6 hpc and between 6 and 18 hpc and the viral load changes significantly between 18 and 26 hpc. In this case, it is not clear exactly how these changes are represented in the simplicial complexes of Fig. 6. Therefore further quantification is required in order to understand in more detail the information that the choice of lenses provide in the biological context.

Mechanistic modeling studies investigating stages of the immune system in response to co-infections have been done previously [4,20,21,32]. While under specific assumptions these models can highlight relevant mechanism, the design of mechanistic models and the abstraction complexity remain largely debatable. Here, TDA is presented as an additional tool to abstract high dimension data sets during co-infections, thereby significantly extending current knowledge and building a basis for translating improved mathematical models into potential therapies.

Declaration of Competing Interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; and in the decision to publish the results.

CRediT authorship contribution statement

Karin Sasaki: Writing - original draft. **Dunja Bruder:** Writing - review & editing. **Esteban A. Hernandez-Vargas:** Conceptualization, Resources, Supervision, Writing - review & editing.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (HE-7707/5-1, BR2221/6-1), and the Alfons und Gertrud Kassel-Stiftung. This work was also supported by the Universidad Nacional Autonoma de Mexico (UNAM) and by the LOEWE CMMS at FIAS. We are also grateful to the authors of the Keppler Mapper Python library who diligently answered our questions on the use of the library.

References

[1] McArdle AJ, Turkova A, Cunnington AJ. When do co-infections matter? Currentopinion ininfectiousdiseases 2018;31(3):209–15. doi:10.1097/QCO.0000000000000447. 29698255[pmid].
[2] Sofonea MT, Alizon S, Michalakis Y. Exposing the diversity of multiple infection patterns. J Theor Biol 2017;419:278–89. doi:10.1016/j.jtbi.2017.02.011.
[3] Graham AL, Cattadori IM, Lloyd-Smith JO, Ferrari MJ, Bjørnstad ON. Transmission consequences of coinfection: cytokines writ large? Trends Parasitol 2007;23(6):284–91. doi:10.1016/j.pt.2007.04.005.
[4] Duvigneau S, Sharma-Chawla N, Boianelli A, Stegemann-Koniszewski S, Nguyen VK, Bruder D, et al. Hierarchical effects of pro-inflammatory cytokines on the post-influenza susceptibility to pneumococcal coinfection. Scientific Reports 2016;6. 37045 EP.
[5] Gartlehner G, Stepper K. Julius Wagner-Jauregg: pyrotherapy, simultanmethode, and 'racial hygiene'. J R Soc Med 2012;105(8):357–9. doi:10.1258/jrsm.2012.12k0049.
[6] Nacher M. Interactions between worms and malaria: good worms or bad worms? Malaria journal 2011;10. doi:10.1186/1475-2875-10-259. 259–259.
[7] Babu S, Nutman TB. Helminth-tuberculosis co-infection: an immunologic perspective. Trends Immunol 2016;37(9):597–607. doi:10.1016/j.it.2016.07.005.
[8] Birger RB, Kouyos RD, Cohen T, Griffiths EC, Huijben S, Mina MJ, et al. The potential impact of coinfection on antimicrobial chemotherapy and drug resistance. Trends Microbiol 2015;23(9):537–44. doi:10.1016/j.tim.2015.05.002.
[9] Mark EJ, Woolhouse RA, Haydon DT. Emerging pathogens: the epidemiology and evolution of species jumps. Trends Ecol Evol 2005;20(5).
[10] Hudson SECIM PJ, Perkins. The emergence of wildlife disease and the application of ecology.. USA: Princeton University Press Princeton; 2008. ISBN 9780691124858.
[11] Hernandez-Vargas EA. Modeling and Control of Infectious Diseases: With MATLAB and R. 1st ed. Elsevier Academic Press; 2019.
[12] Nicolau M, Levine AJ, Carlsson G. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. Proc Natl Acad Sci 2011;108(17):7265–70. doi:10.1073/pnas.1102826108.
[13] Li L, Cheng W-Y, Glicksberg BS, Gottesman O, Tamler R, Chen R, et al. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. Science Translational Medicine 2015;7(311). doi:10.1126/scitranslmed.aaa9364. 311ra174–311ra174.
[14] Bassett DS, Sporns O. Network neuroscience. Nature Neuroscience 2017;20. 353 EP.
[15] Torres BY, Oliveira JHM, Thomas Tate A, Rath P, Cumnock K, Schneider DS. Tracking resilience to infections by mapping disease space. PLoS Biol 2016;14(4):1–19. doi:10.1371/journal.pbio.1002436.
[16] Taylor D, Klimm F, Harrington HA, Kramár M, Mischaikow K, Porter MA, et al. Topological data analysis of contagion maps for examining spreading processes on networks. Nature Communications 2015;6. 7723 EP.
[17] Chan JM, Carlsson G, Rabadan R. Topology of viral evolution. Proc Natl Acad Sci 2013;110(46):18566–71. doi:10.1073/pnas.1313480110.
[18] Cámara PG, Levine AJ, Rabadán R. Inference of ancestral recombination graphs through topological data analysis. PLoS computationalbiology 2016;12(8). doi:10.1371/journal.pcbi.1005071. e1005071–e1005071.
[19] Singh G, Memoli F, Carlsson G. Topological methods for the analysis of high dimensional data sets and 3D object recognition. Eurographics symposium on point-based graphics. Botsch M, Pajarola R, Chen B, Zwicker M, editors. The Eurographics Association; 2007. doi:10.2312/SPBG/SPBG07/091-100.

[20] Smith AM, Adler FR, Ribeiro RM, Gutenkunst RN, McAuley JL, McCullers JA, et al. Kinetics of coinfection with influenza A virus and Streptococcus pneumoniae.. PLoS Pathogens 2013;9(3):e1003238.
[21] Smith AM, Smith AP. A critical, nonlinear threshold dictates bacterial invasion and initial kinetics during influenza. Scientific Reports 2016;6. 38703 EP.
[22] Hernandez-Vargas EA, Boianelli A, Hernandez-Mejia G. Bacterial pneumonia fate decisions. IFAC-PapersOnLine 2018;51(27):390–5. doi:10.1016/j.ifacol.2019.02.001.
[23] Almocera AES, Hernandez-Mejia G, Parra-Rojas C, Hernandez-Vargas EA. The trichotomy of pneumococcal infection outcomes in the host. Commun Nonlinear Sci Numer Simul 2019;73:1–13. doi:10.1016/j.cnsns.2019.01.025.
[24] Nguyen VK, Klawonn F, Mikolajczyk R, Hernandez-Vargas EA. Analysis of practical identifiability of a viral infection model. PLoS One 2016:e0167568.
[25] Saul N., van Veen H.J.. MLWave/Kepler-Mapper: 186f (Version 1.0.1). Zenodo. http://doi.org/10.5281/zenodo.1054444; 2017.
[26] Edelsbrunner H, Harer J. Computational Topology - an Introduction.. Am Math Soc; 2010.
[27] Hatcher A. Algebraic topology. Cambridge: Cambridge Univ. Press; 2000.
[28] Scott D. Multivariate density estimation: theory, practice, and visualization. John Wiley and Sons 1992.
[29] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.
[30] Smith AM. Host-pathogen kinetics during influenza infection and coinfection: insights from predictive modeling. Immunol Rev 2018;285(1):97–112. doi:10.1111/imr.12692.
[31] McCullers JA. Insights into the interaction between influenza virus and pneumococcus. Clin Microbiol Rev 2006;19(3):571–82. doi:10.1128/CMR.00058-05.
[32] Smith AM, McCullers JA, Adler FR. Mathematical model of a three-stage innate immune response to a pneumococcal lung infection. JTheorBiol 2011;276(1):106–16. doi:10.1016/j.jtbi.2011.01.052.